SPeech Phonetization Alignment and Syllabification (SPPAS): a tool for the automatic analysis of speech prosody

Brigitte Bigi, Daniel Hirst

SPeech Phonetization Alignment and Syllabification (SPPAS): a tool for the automatic analysis of speech prosody.

Brigitte Bigi, Daniel Hirst

Laboratoire Parole et Langage, CNRS & Aix-Marseille Université

brigitte.bigi@lpl-aix.fr, daniel.hirst@lpl-aix.fr

Abstract

SPASS, SPeech Phonetization Alignment and Syllabification, is a tool to automatically produce annotations which include utterance, word, syllable and phoneme segmentations from a recorded speech sound and its transcription. SPPAS is currently implemented for French, English, Italian and Chinese and there is a very simple procedure to add other languages. The tool is developed for Unix based platforms (Linux, MaxOS and Cygwin on Windows) and is specifically designed to be used directly by linguists in conjunction with other tools for the automatic analysis of speech prosody. The tools will all be distributed under a GPL license.

Index Terms: phonetic, annotation, speech prosody, intonation

1. Introduction

The last ten years or so has seen an explosion in the quantity of linguistic data which has become available as evidence for the nature of linguistic representations of speech. Whereas it was common some years ago to formulate prosodic models on the basis of rather limited data, today it is becoming more and more expected for linguists to take into account large quantities of empirical data, often including several hours of recorded speech.

Up until a few years ago it was difficult, if not impossible, to manipulate such large quantities of data outside of specialized laboratories using expensive main-frame computers. Software for the analysis of acoustic data was often difficult to install and use, nearly always platform specific and, above all, very expensive.

Today the situation has changed radically and the availability of cheaper and cheaper data storage has made it possible to envisage data analysis involving several hundreds of gigabytes of data. In the same period, a number of software tools for the analysis of audio and/or video recordings of speech have become available such as *Anvil* [1], *Elan* [2], *Praat* [3], *Transcriber* [4] and *WaveSurfer* [5], to name just some of the most popular tools, all of which are both free and multi-platform. For an extensive list of speech analysis software see [6].

The biggest obstacle today facing linguists is not the storage of data, nor its analysis, but its annotation.

2. Tools for prosody analysis

The description of the prosodic structure of speech requires:

> "the explicit characterization of the length, pitch and loudness of the individual sounds which make up an utterance". [7]

Linguists need tools for the automatic analysis of at least these three prosodic features of speech sounds. In this paper we concentrate on the tool for the automatic analysis of segmental duration. We return in the final section to the question of interfacing the tool we describe here with the analysis of other prosodic parameters.

The analysis of the prosodic structure of speech nearly always requires the alignment of the speech recording with a phonetic transcription of the speech, usually in some version of the International Phonetic Alphabet. This task is extremely labour-intensive - it may require several hours for even an experienced phonetician to transcribe and align a single minute of speech manually. It is consequently obvious that transcribing and aligning several hours of speech by hand is not generally something which can be envisaged.

A number of tools are currently available which can be used to automate the task, including the *HTK Toolkit* [8], *Festival* [9], *Julius* [10], the *P2FA* [11], *EasyAlign* [12],

For most of these, our experience is that the tools require a level of expertise in computer science which is beyond the reach of most linguists without the help of an engineer. P2FA, for example, assumes that HTK is installed on the user's computer and it is run via a text-based command-line interface. The *EasyAlign* Praat plugin [12] is currently the most "linguist-friendly" but it currently runs only on Windows and relies on components which cannot all be distributed with a GPL license. EasyAlign is currently available for French and English although the author claims that adding a new language is relatively easy and announces forthcoming versions for German, Dutch and Hebrew.

We are developing a GPL licensed set of tools for Unix based platforms, including Linux, Mac-OSX and Windows (using Cygwin). The tool we describe here implements the automatic phonetization, alignment and syllabification of speech recordings and it is specifically designed to be used directly by linguists in conjunction with other tools for the automatic analysis of speech prosody.

3. SPPAS description

3.1. Overview

SPPAS - SPeech Phonetization Alignment and Syllabification, is a tool to automatically produce annotations which include utterance, word, syllable and phoneme segmentations from a speech recording and its transcription. *SPPAS* is currently implemented for French, En-

glish, Italian and Chinese and there is a very simple procedure to add other languages.

The whole process is a succession of 4 automatic steps. The resulting alignments are a set of TextGrid files, the native file format of the Praat software [3] which has become the most popular tool for phoneticians today. *SPPAS* generates separate TextGrid files for (i) utterance segmentation, (ii) word segmentation, (iii) syllable segmentation and (iv) phoneme segmentation. The output is illustrated as a Screenshot in Figure 1, with TextGrid files merged in Praat. For the Chinese example, only 3 tiers are illustrated with the *tokens* tier corresponding to both syllables and morphemes since the corpus was transcribed in Pinyin.

A important point for software which is intended to be widely distributed is its licensing conditions. *SPPAS* uses only resources, tools and scripts which can be distributed under the terms of the GPL license. *SPPAS* tools and resources are currently available at the URL:

http://www.lpl-aix.fr/∼bigi/sppas/.

Recently, the software was successfully applied during the Evalita 2011 campaign, on Italian map-task dialogues. EVALITA is an initiative devoted to the evaluation of Natural Language Processing and Speech tools for Italian[1]. Systems were required to align audio sequences of spoken dialogues to the provided relative transcriptions. Final results report a correct phoneme alignment rate of 88% (both phonetization and alignment errors), and a correct word alignment rate of 97.6%.

There is a simple procedure to add new languages in *SPPAS*: it requires only putting a dictionary and an acoustic model into the appropriate directory.

3.2. Inter-Pausal Unit Segmentation

Inter-pausal unit segmentationn is a completely open research problem. It consists in aligning sentences of a document with the corresponding sound. Our current implementation of SPPAS includes an algorithm for this step which we hope to improve in future versions of the software.

The algorithm currently implemented identifies silent pauses in the signal and attempts to align them with the inter-pausal units proposed in the transcription, under the assumption that each such unit is separated by a silent pause. For a given minimum duration for pauses and for inter-pausal units a dichotomic search adjusts the silence threshold (in dB) and identifies the number of units thus defined. If the number of units found is greater or less than the desired number, the search is renewed adjusting the minimum duration of the silences and units accordingly. The search halts when the three parameters are correctly adjusted: minimal duration of pauses, minimal duration of units and silence threshold. Silent pauses can be indicated in the transcription either by the symbol '#' or by a newline, or by both together.

A recorded speech file with the *.wav* extension must correspond to each *.txt* file. The correspondences are established by means of the file names.

[1] http://www.evalita.it/

3.3. Phonetization

Clearly, there are different ways to pronounce the same utterance. Different speakers have different accents and tend to speak at different rates. Phonetization is the process of representing sounds by phonetic symbols. There are two general ways to construct a phonetization process: rule based systems (with rules based on inference approaches or proposed by expert linguists) and dictionary based solutions which consist of storing a maximum of phonological knowledge in a lexicon. *SPPAS* implements both approaches for French, but currently only the second for other languages since the rule-based approach is of course highly language dependent.

The program for the phonetization of the orthographic transcription produces a phonetic transcription based on a phonetic dictionary. In the case of phonetic variants, no rules are applied: all possibilities are taken into consideration. The phonetization contains all possible grapheme-to-phoneme conversions of each word of the transcription. If a word cannot be found in the dictionary, the phonetization is "UNK". In this case, the utterance containing the missing word will not be aligned: the missing word will have to be added to the dictionary and *SPPAS* will need to be re-started.

To perform the phonetization, an important step is to build the pronunciation dictionary, where each word in the vocabulary is expanded into its constituent phones. The phonetization is the equivalent of a sequence of dictionary-look-ups. It supposes that all words of the speech transcription are mentioned in the pronunciation dictionary. Actually, some words can correspond to several entries in the dictionary with various pronunciations. To deal with such a case, *SPPAS* decides on the correct pronunciation during the alignment step because the pronunciation can generally be inferred from the speech signal.

For example, the following input sentence: "I never get to sleep on the airplane" will produce the following phonetization: "ay n.eh.v.er g.eh.t|g.ih.t t.uw|t.ix|t.ax s.l.iy.p aa.n|ao.n dh.ax|dh.ah|dh.iy eh.r.p.l.ey.n" using the Arpabet transcription alphabet.

In this phonetization, spaces separate words, dots separate phonemes and pipes (|) separate phonetic variants. The user can either let the system choose the correct phonetization automatically or can force the transcription manually by selecting the appropriate pronunciation of the sentence.

3.4. Alignment

Phonetic alignment consists in a time-matching between a speech utterance and a phonetic representation of the utterance. *SPPAS* is based on the *Julius Speech Recognition Engine* (SRE) [10]. For each utterance, the orthographic and phonetic transcriptions are used. Julius performs an alignment to identify the temporal boundaries of phones and words. Julius was originally designed for dictation applications, and its distribution includes only Japanese acoustic models. Since it uses acoustic models trained using the HTK toolkit [8], however, it can also use acoustic models trained for other languages.

To perform alignment, a finite state grammar describing sentence patterns to be recognized and an acoustic model are needed. A grammar essentially defines con-

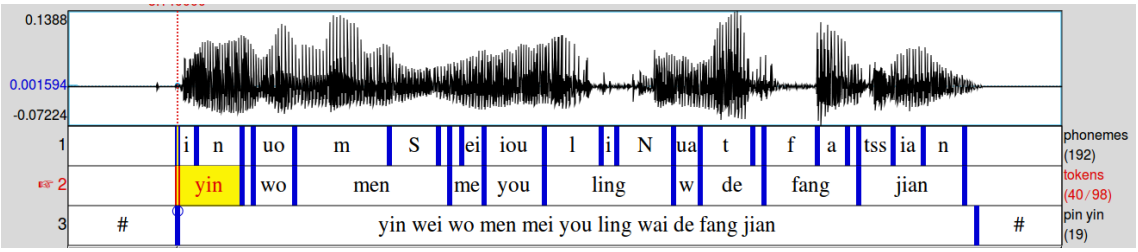

Figure 1: SPPAS *output example for the utterance* **. Pinyin: Yīnwèi wǒmen méiyǒu lìngwài de fángjiān (Because we do not have another room)**

straints on what can be expected as input. It is a list of words that the SRE listens for. Each word has a list of associated phonemes, taken from the dictionary. For a speech input, Julius searches for the most likely word sequence under the constraints of the given grammar.

Speech alignment also requires an acoustic model in order to align speech. An acoustic model is a file that contains statistical representations of each of the distinct sounds of one language. Each phoneme is represented by one of these statistical representations. *SPPAS* is based on the use of HTK-ASCII acoustic models. Acoustic models were trained with HTK by taking the training corpus of speech, previously segmented into utterances and phonetized. HTK (Hidden Markov Toolkit) is a portable toolkit for building and manipulating the statistical Hidden Markov models (HMMs) used to represent sound in Speech Recognition. HTK's licence requires registering before it can be downloaded. The software is open source but there are limitations on the distribution of the HTK Toolkit itself. For example, *an HTK tool should not be embedded in another tool*. There is, however, no limitation on the distribution of the models created with the toolkit, which can consequently be distributed under GPL license.

In the example presented in the previous section, the system chose the following phonetization: "ay n.eh.v.er g.eh.t t.ax s.l.iy.p aa.n dh.ax eh.r.p.l.ey.n".

3.5. Syllabification

The syllabification of phonemes is performed with a rule-based system previously described for French in [13]. A new set of rules was developed to deal with Italian and English. In the previous example, the system proposed the following syllabification (spaces separate syllables): "ay n.eh v.er g.eh.t t.ax s.l.iy p aa.n dh.ax eh.r p.l.ey.n".

4. Resources

4.1. Phonetic resources

An important step is building pronunciation dictionaries, where each word in the vocabulary is expanded into its constituent phones.

The Italian dictionary was downloaded from the Festival synthesizer tool [14]. This dictionary was enriched by word pronunciations observed in the Evalita training corpus. We manually corrected a large set of both these phonetizations. The final version of the dictionary contains about 390k words and 5k variants.

The French dictionary was constructed by merging several available dictionaries. Some word pronunciations were also added using the LIA_Phon tool. It contains about 350k words and 300k variants.

The English dictionary that *SPPAS* uses is taken from the VoxForge project [15], under the terms of the GPL license.

The Chinese dictionary was hand-made. It now contains about 350 tokens and is still in progress. While this number may seem small compared to that of the other languages it should be noted that the inventory of syllables in Standard Chinese is generally held to consist of only about 410 phonemically distinct syllables, not counting tones. The Chinese alignment was carried out on a corpus transcribed with the Pinyin system. The present version does not carry out conversion from Chinese characters to Pinyin but it would be very simple to incorporate such a resource.

4.2. Acoustic Model Training

SPPAS is based on common practice and uses context-independent HMMs for speech segmentation. *SPPAS* can use various types of acoustic models. This section describes how we created models which are included in *SPPAS*. The Italian, French and Chinese models were trained following these steps. An English model can be downloaded from the VoxForge web site.

The phoneme statistical representation is based on a 5-state model with a left-to-right topology with self-loops and no transitions skipping over states

Ideally, the phones would have unique articulatory and acoustic correlates. But the acoustic properties of a given phone can depend on the phonetic environment. This phenomenon of coarticulation motivated the adoption of context-dependent models such as triphones.

Our training procedure was adapted from the VoxForge tutorial. Typically, the HMM states are modeled by Gaussian mixture densities whose parameters are estimated using an expectation maximization (EM) procedure. The outcome of this training procedure is dependent on the availability of accurately annotated data and on good initialization. As more audio speech data is collected, better acoustic models can be created. Acoustic models were trained from 16 bit, 16000 Hz wav files. The Mel-frequency cepstrum coefficients (MFCC), along with their first and second derivatives, were extracted from the speech in the standard way.

The acoustic model training procedure comprises 3

main steps. Step 1 is data preparation. It establishes the list of phonemes, plus silence and short pauses. It converts the input data (phonetization) into the HTK-specific data format. It codes the (audio) data (= "parameterizing the raw speech waveforms into sequences of feature vectors") from wav format to MFCC. Step 2 is the generation of monophones. It creates a Flat Start Monophone model by defining a prototype model and copying this for each phoneme. Next, this flat model is re-estimated using the MFCC files to create a new model. Then, it fixes the "sp" model from the "sil" model by extracting only 3 of the initial 5 model states. This model is then re-estimated using the MFCC files and the phonetization. Step 3 creates tied-state triphones from monophones and from some language specificities defined by means of a configuration file. This file summarizes phonemic informations such as for example the list of vowels, liquids, fricatives, nasals or stop. We created this resource manually for Italian, French and Chinese.

4.3. Adding a new language to *SPPAS*

SPPAS can deal with a new language L by simply adding the language resources to the appropriate directories:

- the dictionary to: SPPAS/dict/L.dict
- the acoustic model to: SPPAS/models-L

The only step in the procedure which is probably beyond the means of a linguist without external aid is the creation of a new acoustic model when it does not already exist for the language being analyzed. This only needs to carried out once for each language, though, and we plan to provide detailed specifications of the information needed to train an acoustic model on an appropriate set of recordings and dictionaries or transcriptions. Acoustic models obtained by such a collaborative process will be made freely available to the scientific community.

5. Interfacing with other tools

There are today a number of automatic systems for the automatic modeling and coding of pitch including *Momel/INTSINT* [16, 17], which was developed in our lab.

SPPAS and Momel have a number of characteristics in common. Both can be used entirely automatically, with of course the risk of error that that implies, but also the possibility of applying the programs to very large quantities of data. They can also both be run in a supervised mode which allows the user to intervene and to manually correct each intermediate step. Both systems have, moreover, been conceived as language-independent modules - for SPASS only the resources are language dependent, not the tools or the algorithms.

Our current undertaking is to provide these two programs with a common interface so that they can be used together. At present there is, to our knowledge, no generally available algorithm modeling the *loudness* of speech sounds in a useful way but when and if such algorithms become available, we will of course be pleased to integrate them into our speech analysis environment.

6. Conclusions

SPPAS is a tool to perform automatic phonetization, alignment and syllabification. Current development is in progress to improve the portability and the accessibility (a GUI is still under development). *SPPAS* is specifically designed with the aim of providing a tool not for computer-scientists but rather for phoneticians because no such tool is currently available under a GPL licence.

7. Acknowledgements.

Our thanks to Masahiko Komatsu for allowing us to use the Chinese Multext corpus [18] and to Na Zhi for her help in developing the resources for annotating Chinese.

8. References

[1] M. Kipp., "Anvil, DFKI, German Research Center for Artificial Intelligence," http://www.anvil-software.de/, 2011.

[2] "ELAN - Linguistic annotator. Language archiving technology portal [Computer Software]. Nijmegen: Max Planck Institute for Psycholinguistics. ," http://www.lat-mpi.eu/tools/elan/.

[3] P. Boersma and D. Weenink, "Praat: Doing phonetics by computer. [Computer Software] Amsterdam: Department of Language and Literature, University of Amsterdam. ," http://www.praat.org/, 2011.

[4] "TranscriberAG. A tool for segmenting, labeling and transcribing speech. [Computer Software] Paris: DGA," http://transag.sourceforge.net/, 2011.

[5] "WaveSurfer." http://www.speech.kth.se/wavesurfer/.

[6] J. Llisterri, "Speech analysis and transcription software." http://liceu.uab.es/~joaquim/phonetics/fon_anal_acus/herram_anal_acus.html, 2011.

[7] D. J. Hirst, "ProZed: A speech prosody analysis-by-synthesis tool for linguists." in *Proceedings of the 6th International Conference on Speech Prosody.*, Shanghai, 2012 (submitted).

[8] S. Young and S. Young, "The HTK Hidden Markov Model Toolkit: Design and Philosophy," *Entropic Cambridge Research Laboratory, Ltd*, vol. 2, pp. 2–44, 1994.

[9] The Centre for Speech Technology Research, "The festival speech synthesis system," 2011.

[10] Nagoya Institute of Technology, "Open-Source Large Vocabulary CSR Engine Julius, rev. 4.1.5," 2010.

[11] J. Yuan and M. Liberman, "Speaker identification on the scotus corpus," in *Proceedings of Acoustics 2008*, 2008, pp. 5687–5690.

[12] J.-P. Goldman, "EasyAlign: a friendly automatic phonetic alignment tool under Praat," in *Proceedings of Interspeech XI.*, no. Ses1-S3:2, Florence, Italy, 2011.

[13] B. Bigi, C. Meunier, I. Nesterenko, and R. Bertrand, "Automatic detection of syllable boundaries in spontaneous speech," in *Language Resource and Evaluation Conference*, La Valetta, Malta, 2010, pp. 3285–3292.

[14] A. W. Black and P. A. Taylor, "The Festival Speech Synthesis System," Human Communciation Research Centre, University of Edinburgh, Scotland, UK, Tech. Rep. HCRC/TR-83 [Avaliable from http://www.cstr.ed.ac.uk/projects/festival.html.], 1997.

[15] "Voxforge," http://www.voxforge.org, 2006-2011.

[16] D. J. Hirst and R. Espesser, "Automatic modelling of fundamental frequency using a quadratic spline function." *Travaux de l'Institut de Phonétique d'Aix*, vol. 15, pp. 75–85, 1993.

[17] D. J. Hirst, "The analysis by synthesis of speech melody: from data to models." *Journal of Speech Sciences*, vol. 1, no. 1, pp. 55–83, 2011.

[18] M. Komatsu, "Chinese MULTEXT: Recordings for a Prosodic Corpus," *Sophia Linguistica*, vol. 57, 2010.